```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

[1] ✓ 2.5s Python

Reading Dataset

+ Code + Markdown

```
data = pd.read_csv("C:/Users/SANDY/Desktop/DATA ANALYSIS FILES - JULY 2024/CognoRise DS Internship Oct 2024/1. IRIS DATASET PROJECT/IRIS.csv")
```

[2] ✓ 0.0s Python

```
data.head()
```

[3] ✓ 0.0s Python

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
data.describe()
```

[4] ✓ 0.0s Python

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000

```
data.info()
```

[5] ✓ 0.0s Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

Checking for NULL Values

```
data.isnull().sum()
```

[6] ✓ 0.0s Python

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

Data Preprocessing

```
X = data.drop(columns=['species'])
y = data['species']
```

[7] ✓ 0.0s Python

Splitting the DATASET

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

[14] ✓ 0.0s Python

Feature Scaling

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

[15] ✓ 0.0s Python

Model Selection and Training (Logistic Regression)

```
model = LogisticRegression()
model.fit(X_train, y_train)
```

[16] ✓ 0.0s Python

```
LogisticRegression()
```

Predictions & Evaluations

```
y_pred = model.predict(X_test)
y_pred
```

[17] ✓ 0.0s Python

```
array(['Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-virginica',
       'Iris-setosa', 'Iris-virginica', 'Iris-setosa', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-setosa', 'Iris-setosa'], dtype=object)
```

```
print("Accuracy: ", accuracy_score(y_test, y_pred))
```

[18] ✓ 0.0s Python

```
Accuracy:  1.0
```

```
Accuracy:  1.0
```

```
classification_report(y_test, y_pred)
```

[19] ✓ 0.0s Python

```
'              precision    recall  f1-score   support\n\n    Iris-setosa       1.00      1.00      1.00        10\nIris-versicolor       1.00      1.00      1.00         9\n Iris-virginica       1.00
```

```
confusion_matrix(y_test, y_pred)
```

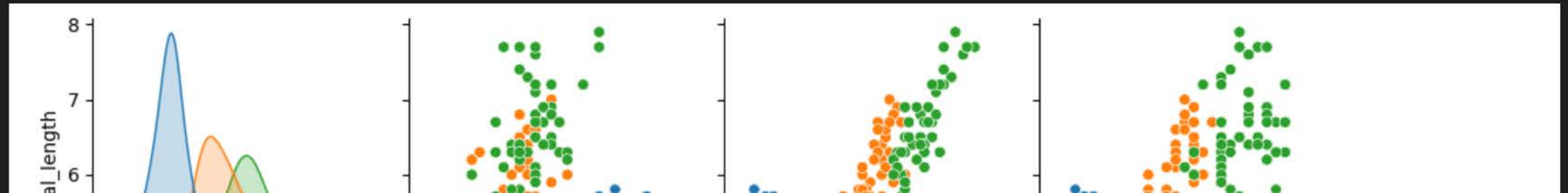
[20] ✓ 0.0s Python

```
array([[10,  0,  0],
       [ 0,  9,  0],
       [ 0,  0, 11]])
```

Visualization

```
sns.pairplot(data, hue="species")
plt.show()
```

[21] ✓ 2.3s Python



```
       [ 0,  0, 11]])
```

Visualization

```
sns.pairplot(data, hue="species")
plt.show()
```

[21] ✓ 2.3s Python